

Operating System Level

- Provides layers of abstraction in the form of system calls
 - Reading/writing files
 - Sending/receiving network data
 - Using other I/O devices - mice, keyboards, video card, sound card, etc.
 - Executing programs
- Scheduling processes
- Manages memory

Process

- An instance of an executing program
- Items involved:
 - Memory
 - Registers
 - I/O status (what files are open, whether it's waiting for input, etc)
 - Permissions - what user is running the process
 - etc.

Program vs Process

- A program is stored, it does not do anything on its own
- A process is an instance of a program in execution
- There can be multiple processes running the same program at the same time
- A program can be changed or deleted while a process is running that program and process is unaffected

Memory

- A programmer's ideal memory
 - Infinitely large
 - Infinitely fast
 - Private
 - Non volatile
 - Cheap
- Settle for a memory hierarchy
 - Registers
 - Cache
 - Main memory
 - SSDs
 - Hard disks
 - Removable storage

- Early computers

- No memory abstraction

- A single process accessed physical memory by address, the process could access any address

- Allowing multiple processes in memory requires:

- Protection - a process must not be able to access another process's memory

- Relocation - the OS must be able to move a process's memory to another part of RAM or even to disk without the process being aware

Address Space

- Set of addresses a process can use to address memory
- Each process has its own independent address space
- Virtual memory
 - Maps addresses from a process's address space to physical addresses
 - Physical memory can be in main memory or on disk
 - Memory can be moved from main memory to disk as needed
 - If main memory is full and more memory is needed, move idle memory to disk

- Programs access memory using virtual addresses that map to physical addresses
- Memory Management Unit (MMU)
 - Maps virtual addresses to physical addresses
 - Usually on the CPU